

REMARKS/ARGUMENTS

Claims 1, 3-6, 11-18, and 20-26 are pending in the application and have been rejected. Claims 2, 7-10, 19 and 27 have been cancelled and claims 1, 11, 18 have been amended.

Rejections under 35 U.S.C. §103

The Office Action rejected claims 1-9, and 18-27 as unpatentable over United States Patent No. 6,304,949 to Houlsdworth in view of United States Patent 5,761,670 to Joy.

Claim 1 recites a method of managing memory in a multi-threaded processing environment including local thread stacks and local thread heaps, and a global heap. The method, as amended, comprises the following steps or acts:

A method of managing memory in a multi-threaded processing environment including respective local thread stacks and heaps and a global heap, said method comprising:

creating an object in a thread heap;

monitoring the object to determine whether the object is referenced only from a given thread stack;

associating a local status with the object when the object is referenced only from a given thread stack;

changing the status of the object to a global status when the object is not referenced from only the given thread stack; and

deleting from a given thread heap one or more local objects when the object is associated with a local status.

Houlsdworth relates to a data processing apparatus for handling multi-thread programs. The Office Action concedes that Houlsdworth does not disclose monitoring the object to determine whether the object is referenced only from a given thread stack and changing the status of the object under certain conditions.

Houlsdworth also fails to teach or suggest the claimed "creating an object in a thread heap" -- the passages cited by the Office Action from Houlsdworth discuss a single heap for all of the objects (in particular lines 13-15 of column 4). We partition the heap into sections for the use of individual threads. These are the thread heaps mentioned in the claims. Houlsdworth says

nothing about such thread heaps. Therefore, Houlsdworth neither teaches nor suggests the “creating an object in a thread heap” step.

Applicant adds that Houlsdworth further fails teach or suggest the claim limitation of “changing the status of the object to a global status when the object is not referenced from only the given thread stack.”

The Office Action contends that Joy teaches the elements of “monitoring the object” and “changing the status” and that it would have been obvious to one skilled in the art to modify Houlsdworth according to Joy. Applicant respectfully submits that the claims at issue would not have been obvious in view of the combination of Houlsdworth and Joy.

We claim objects that are local or global; Joy discusses locking procedures that are local and global. Joy’s global procedures, e.g., locking procedure, work for all objects. Joy’s local procedure, e.g., local locking procedure is a procedure that has been specialized for a particular object. Joy’s procedures have nothing to do with the claimed accessibility or thread-locality, and thus are totally unrelated to the claimed subject matter. Thus, there is no way to do dynamic tracking of locality/globality as claimed based on what Joy teaches.

Moreover, Joy neither teaches nor suggests “monitoring” as claimed. Joy mentions monitors as a way to synchronize access to objects. The word “monitor” is used in the claims in a totally different way. The meaning of “monitor” as used in the claims is the usual English language plain meaning: “to check, watch or keep track of.” The claimed method monitors each object to determine the accessibility of the object from any thread other than the given thread (i.e., thread locality). The cited passage from Joy (col. 1, lines 43-52) relates to determining ownership of an object, not accessibility from other threads as claimed.

The cited references also fail to teach or suggest the claimed “changing the status of the object.” The passages cited from Joy by the Office Action discuss changing the status of an object from “locked” to “unlocked” or vice versa (as a result of a program locking or unlocking an object). This is different from changing the status of an object from local to global based on accessibility or reachability (i.e., as a result of updates to objects or variables).

Moreover, the Office Action does not provide any evidence of a motivation for combining the references. Certainly, the Office Action rationale for making the combination is

neither taught nor suggested by Joy or the combination of the cited references.

There is no mechanism in Joy for tracking the stack locality of objects as claimed; Joy specifically states the rules which govern the placement of objects in the stack, thread local heap and global heap. The placement decision is made once and not subsequently altered. The claims by contrast specifically recite a mechanism by which an object can be assigned to a local heap and then moved to the global heap when it becomes globally accessible.

Therefore, those skilled in the art would not have been motivated by either of the cited references, whether viewed individually or in combination, to make the claimed invention.

Claims 18 and 22 are respectively machine and article of manufacture counterparts of claim 1 and those claims as well as their dependent claims are patentable for the reasons discussed above.

The Office Action contends that the limitation of claim 5 (changing the status of an object in the thread heap to global if the object is assigned to a static variable or if the object is assigned to a field in any other object) is disclosed at Houlsdworth col. 5, lines 4-20 and 52-56. As discussed, above Houlsdworth neither teaches nor suggests changing the local/global status of an object. Moreover, claim 5 is distinguishable over the cited references for the same reasons that claim 1 is patentable.

The Office Action contends that the limitation of claim 6 (intercepting assignment operations to an object in the thread heap to assess whether the status should be changed) is disclosed by the abstract of Joy. As discussed, above Houlsdworth neither teaches nor suggests changing the local/global status of an object. Moreover, claim 6 is distinguishable over the cited references for the same reasons that claim 1 is patentable.

Claims 7-9 have been cancelled and hence their rejections are moot.

The Office Action rejected claims 10-13 as unpatentable over United States Patent No. 6,304,949 to Houlsdworth in view of United States Patent 5,761,670 to Joy and further in view of United States Patent 6,351,794 to Spilo.

The Office Action contends that the limitation of claim 11 (using multiples of 2 or more bytes in a thread heap to store the objects) is disclosed at Spilo col. 5, line 45 – col. 6, line 2. However, the Office Action does not even allege that the limitation “whereby there is at least one

spare bit in the object length variable and using the at least one spare bit as the status" is disclosed anywhere in the cited references. Moreover, claim 11 is distinguishable over the cited references for the same reasons that claim 1 is patentable.

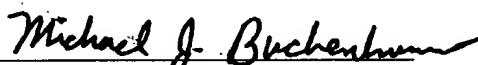
The Office Action contends that the limitation of claim 12 (moving objects whose status is global from the thread heap to the global heap) is disclosed at Spilo Fig. 5, and col.11, lines 33-39. Applicant respectfully traverses this statement. Spilo neither teaches nor suggests this limitation. The cited portion discusses moving half of the TCBs in the local pool to the global pool when the local pool overflows. That has nothing to do with the claimed limitation.

The Office Action contends that the limitation of claim 13 (compacting the reachable local objects in a thread heap) is disclosed at Houlsdworth col. 2, lines 49-65. Applicant respectfully traverses this statement. That section discusses compacting memory space by moving objects it finds to another region. It says nothing about the thread heap.

Claims 14-17 are patentable for the reasons discussed above.

For the foregoing reasons, Applicant respectfully requests allowance of the pending claims and that a timely Notice of Allowance be issued in this case.

Respectfully submitted,


Michael J. Buchenhorner
Reg. No. 33,162

Date: May 12, 2004

HOLLAND & KNIGHT LLP
Holland & Knight LLP
701 Brickell Avenue, Suite 3000
Miami, FL 33131
(305) 789-7773 (voice)
(305) 789-7799 (fax)

1933727_v1